Telnet

What is Telnet?

- The Telnet protocol is often thought of as simply providing a facility for remote logins to computer via the Internet.
- A simple terminal using the local telnet program (known as the client program)
- A Protocol (and a program) that lets you use the power of the Internet to connect you to databases, library catalogs, and other information resources around the world.
- Used in other protocols
  - **FTP Control Connection uses Telnet**




Basic Uses of Telnet

- Telnet lets you connect directly to another computer on the Internet and run programs on the computer (provided you have access permission
  - Username and
  - Password
- Theoretically you have a lot of power at your fingertips.
  - Your dumb terminal or
  - very old PC

Can (potentially) run programs on the worlds most powerful computers.




Problems Using of Telnet?

- Life is not that simple though:
  - Telnet is old. It is text based only.
  - You have to know how to run programs at the other end.
    - Different Computer System — E.g. Unix
    - Different Commands, Directory Structures
    - Different Language — Programming and "spoken" — E.g. Unix Shell Scripts



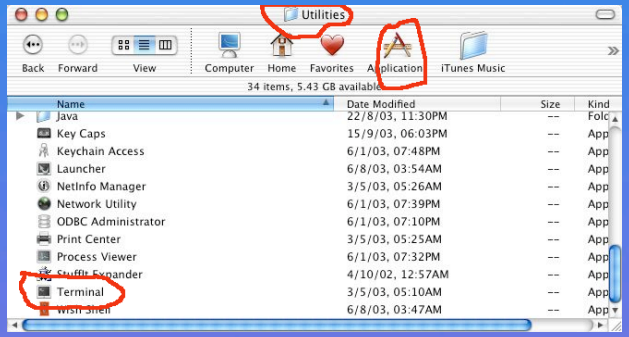

Using Mac Terminal

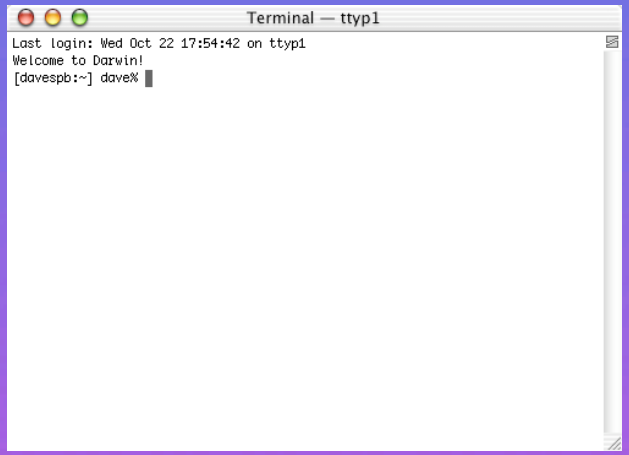
The preferred method of TELNET for this course is to use **Mac Terminal Client** on the Macintosh Computers.

The application is found the in **Applications/Utilities** sub folder.

It essentially fires up a simple command line terminal window.

As far as a user is concerned you do not see any of TELNET protocol communication.





Shells and Secure Shells

You may wish to run applications on remote machines from the command line

- All dialogs with remote machine use TELNET protocol.

For local remote connections to School Computers you can use

`rlogin` **localhost** — to remote login to a given localhost

For External remote connections to Dept. Computers you can use

`ssh` **host** — to secure shell to a given host. Secure shell uses encryption of all TELNET transactions.

For more information of TELNET/Secure Shell on Schools computers see Web or PDF files online.




How Telnet Works:The Network Virtual Terminal

- Communication via TCP/IP protocols
- communication based on a set of facilities known as a Network Virtual Terminal (NVT).
- Telnet client program:
  - responsible for mapping incoming NVT codes to the actual codes needed for user's display device
  - responsible for mapping user generated keyboard sequences into NVT sequences.
- NVT uses 7 bit codes for characters (ASCII Codes),
- The display device, referred to as a printer, is only
- NVT ASCII is used by many other Internet protocols.




NVT Control Codes

The following control codes are required to be understood by the Network Virtual Terminal.

Name	code	Decimal Value	Function
NULL	NUL	0	No operation
Line Feed	LF	10	Moves the printer to the next print line, keeping the same horizontal position
Carriage Return	CR	13	Moves the printer to the left margin of the current line




Back
Close

NVT Control Codes (Cont.)

The following further control codes are optional but should have the indicated defined effect on the display.

Name	code	Decimal Value	Function
BELL	BEL	7	Produces an audible or visible signal (which does NOT move the print head)
Back Space	BS	8	Moves the print head one character position towards the left margin. On a printing devices this mechanism was commonly used to form composite characters by printing two basic characters on top of each other.
Horizontal Tab	HT	9	Moves the printer to the next horizontal tab stop. It remains unspecified how either party determines or establishes where such tab stops are located.




NVT Control Codes (Cont.)

Vertical Tab	VT	11	Moves the printer to the next vertical tab stop. It remains unspecified how either party determines or establishes where such tab stops are located.
Form Feed	FF	12	Moves the printer to the top of the next page, keeping the same horizontal position. On visual displays this commonly clears the screen and moves the cursor to the top left corner.




Back
Close

Telnet Commands

Name	Decimal Code	Meaning
SE	240	End of subnegotiation parameters.
NOP	241	No operation
DM	242	Data mark. Indicates the position of a Synch event within the data stream. This should always be accompanied by a TCP urgent notification.
BRK	243	Break. Indicates that the "break" or "attention" key was hit.
IP	244	Suspend, interrupt or abort the process to which the NVT is connected.
AO	245	Abort output. Allows the current process to run to completion but do not send its output to the user.
AYT	246	Are you there? Send back to the NVT some visible evidence that the AYT was received.




Telnet Commands (Cont.)

EC	247	Erase character. The receiver should delete the last preceding undeleted character from the data stream.
EL	248	Erase line. Delete characters from the data stream back to but not including the previous CRLF.
GA	249	Go ahead. Used, under certain circumstances, to tell the other end that it can transmit.
SB	250	Subnegotiation of the indicated option follows.
WILL	251	Indicates the desire to begin performing, or confirmation that you are now performing, the indicated option.
WONT	252	Indicates the refusal to perform, or continue performing, the indicated option.




Back
Close

Telnet Commands (Cont.)

DO	253	Indicates the request that the other party perform, or confirmation that you are expecting the other party to perform, the indicated option.
DONT	254	Indicates the demand that the other party stop performing, or confirmation that you are no longer expecting the other party to perform, the indicated option.
IAC	255	Interpret as command




Telnet Options

A variety of options that can be negotiated between a telnet client and server using commands at any stage during the connection.

The following are the most important:

Decimal code	Name
1	echo
3	suppress go ahead
5	status
6	timing mark
24	terminal type
31	window size
32	terminal speed
33	remote flow control
34	linemode
36	environment variables




Telnet Option Negotiation

Options are agreed by a process of negotiation.

Either end of a telnet dialogue can enable or disable an option either locally or remotely. The initiator sends a 3 byte command of the form

```
IAC,<type of operation>,<option>
```

The response is of the same form.

Operation is one of:

Description	Decimal Code	Action
WILL	251	Sender wants to do something.
DO	252	Sender wants the other end to do something.
WONT	253	Sender doesn't want to do something.
DONT	254	Sender wants the other not to do something.




Telnet Responses

Associated with each of the these there are various possible responses :

Sender Sent	Receiver Responds	Implication
WILL	DO	The sender would like to use a certain facility if the receiver can handle it. Option is now in effect
WILL	DONT	Receiver says it cannot support the option. Option is not in effect.
DO	WILL	The sender says it can handle traffic from the sender if the sender wishes to use a certain option. Option is now in effect.
DO	WONT	Receiver says it cannot support the option. Option is not in effect.
WONT	DONT	Option disabled. DONT is only valid response.
DONT	WONT	Option disabled. WONT is only valid response.




Back
Close

Telnet Example

For example if the sender wants the other end to suppress go-ahead it would send the byte sequence

```
255(IAC),251(WILL),3
```

The final byte of the three byte sequence identifies the required action.




Telnet Communication

For some of the negotiable options values need to be communicated once support of the option has been agreed.

This is done using sub-option negotiation.

Values are communicated via an exchange of value query commands and responses in the following form.

```
IAC,SB,<option code number>,1,IAC,SE
```

and

```
IAC,SB,<option code>,0,<value>,IAC,SE
```





For example if the client wishes to identify the terminal type to the server the following exchange might take place

```
Client   255(IAC),251(WILL),24
Server   255(IAC),253(DO),24
Server   255(IAC),250(SB),24,1,255(IAC),240(SE)
Client   255(IAC),250(SB),24,0,'V','T','2','2','0',
              255(IAC),240(SE)
```

The above works as follows:

- The first exchange establishes that terminal type (option number 24) will be handled, the server then enquires of the client what value it wishes to associate with the terminal type.
- The sequence `SB,24,1` implies sub-option negotiation for option type 24, value required (1).
- The `IAC,SE` sequence indicates the end of this request.
- The repsonse `IAC,SB,24,0,'V'...` implies sub-option negotiation for option type 24, value supplied (0), the `IAC,SE` sequence indicates the end of the response (and the supplied value).